02908.000005. PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	)	
	:	Examiner: Philip C. Lee
SEBASTIEN JEAN, et al.	)	
	:	Group Art Unit: 2152
Application No.: 09/853,767	)	
	:	Technology Center: 2100
Filed: May 14, 2001	)	
	:	
For: NETWORK DEVICE MIMIC SUPPORT	)	
	:	November 21, 2006

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to:Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on

November 21, 2006
(Date of Deposit)

Michael K. O'Neill (Reg. No. 32,622)
(Name of Attorney for Applicant)

Signature

November 21, 2006
Date of Signature

APPELLANTS' BRIEF ON APPEAL

Sir:

This Brief is submitted in support of Appellants' appeal from the final rejection of Claims 1 to 40 in the above-identified application. A timely Notice of Appeal was filed with a Petition for Extension of Time on August 21, 2006, along with a Pre-Appeal Brief Request for Review. This Appeal Brief is being filed in accordance with the instruction to proceed to appeal in the Notice of Panel Decision From Pre-Appeal Brief Review. In compliance with 37 CFR § 41.20, submitted herewith is a check in payment of the $500.00 brief fee. A Petition For Extension Of Time with fee is being filed herewith to extend the period for filing the Appeal Brief from October 21, 2006 to November 21, 2006.

11/27/2006 SDENBOB1 00000037 09853767
01 FC:1402 500.00 OP

TABLE OF CONTENTS

TABLE OF AUTHORITIES

(1) REAL PARTY IN INTEREST

The real party in interest herein is the assignee in the present application, Canon Development Americas, Inc., a subsidiary of Canon U.S.A., Inc. Appellants note that Canon Development Americas, Inc., the present assignee, formerly did business as Canon Information Systems, Inc., the assignee of record.

(2) RELATED APPEALS AND INTERFERENCES

Appellants, Appellants' legal representative, and the Assignee are not aware of any other related appeals or interferences which will directly affect, be directly affected by, or have a bearing on the Board's decision in the instant appeal.

(3) STATUS OF CLAIMS

Claims 1 to 40 are pending, of which Claims 1 and 33 are independent. All claims have been rejected finally under 35 U.S.C. § 103(a), and all such rejections are being appealed.

Independent Claim 1 has been finally rejected under 35 U.S.C. § 103(a) over U.S. Application Publication No. 2002/0080391 (Sugiura) in view of U.S. Patent No. 6,816,270 (Cooper).

Independent Claim 33 has been finally rejected under 35 U.S.C. § 103(a) over Sugiura and Cooper in view of U.S. Patent No. 6,611,863 (Banginwar).

The remaining claims are all dependent, and have been finally rejected under 35 U.S.C. § 103(a) as above, or further in view of one or more of the following: U.S. Patent No. 6,240,456 (Teng), U.S. Patent No. 6,757,280 (Wilson), U.S. Patent No. 6,157,950 (Krishnan), U.S. Patent No. 6,020,973 (Levine), and U.S. Patent No. 6,742,039 (Remer).

(4) STATUS OF AMENDMENTS

The claims have not been amended subsequent to the final rejection. The language of the claims is therefore identical to that set forth in the Amendment and Statement of Summary of Interview dated January 20, 2006. In addition, a copy of the claims involved in the appeal is provided in the attached Claims Appendix.

(5) SUMMARY OF CLAIMED SUBJECT MATTER

In accordance with MPEP § 1205.02, elements recited in the claims are identified below with corresponding exemplary elements described in the specification. However, it should be understood that the claimed elements are not limited to the embodiment discussed below, or to the embodiments described in the specification.

Appellants' invention concerns a method for mimicking network devices. A representative embodiment of a network environment in which Appellants' invention may be practiced is shown in Figure 1, which is reproduced below.

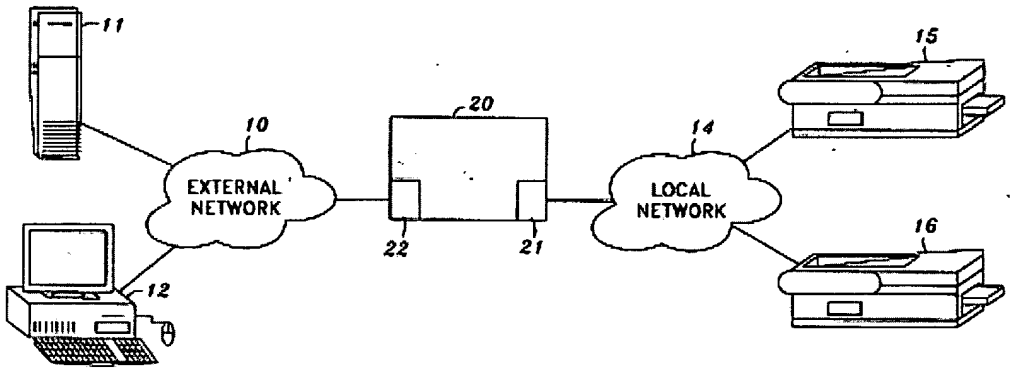

FIG. 1

In Figure 1, device 20 is a mimic device which performs the method of the invention. Mimic device 20 interfaces between external network 10 via network interface card 22 and local network 14 via network interface card 21. The local network 14 includes legacy or other target devices such as printers 15 and 16. The external network 10 includes client devices such as network server 11 and workstation 12, which seek access to functionality offered by the target devices.

One purpose of the invention is to provide added or improved functionality over that provided inherently in the target device itself. For example, the target device might be a legacy device that lacks a more modern functionality, such as a legacy printer that lacks enterprise printing functionality like secure printing or e-mail printing. For this purpose, mimic device 20 acts as a middle-man between external network 10 and local network 14, and acts transparently on behalf of target devices in response to requests for functionality that the target devices might not inherently support.

As claimed, the method is performed in a computing device, such as mimic device 20, having first and second network interface cards, such as first network interface card 22 connecting the computing device to an external network 10, and second network interface card 21 connecting the computing device to a local network 14. (See Specification, page 12, line 31 to page 13, line 6).

In the Figure 1 embodiment, mimic device 20 receives an incoming message from a client network device residing on the external network 10, such as server 11 or workstation 12. The incoming message is addressed to a network address of a target network device residing on the local network 14. That is, the incoming message is addressed to the network address of legacy network printer 15 or legacy network printer 16. (See Specification, page 6, lines 27 to 32 and page 30, line 7 to page 31, line 32).

Thus, while the client network device on the external network sends a message addressed to the network address of a target device on the local network, it is mimic device 20 which actually receives the message. Specifically, first network interface card 22 and second network interface card 21 allow mimic device 20 to act as a "controlled bridge" receiving messages between external network 10 and local network 14. (See

Specification, page 12, line 31 to page 13, line 2). In this way, mimic device is able to "act[] on behalf of legacy network devices by responding to network messages addressed to the legacy network devices." (Specification, page 44, lines 17 to 20; See also Specification, page 3, lines 27 to 30).

In one example, these messages are addressed to the IP address of a target device. In representative examples from the specification, mimic device 20 "receiv[es] an incoming message from a client network device residing on the external network, the message being directed to an IP address of a designated one of the plurality of legacy network printers." (Specification, page 6, lines 27 to 32). In another example, "the mimic device intercepts requests from a client on the external network for e-mail printing from a network printer on the local network.". (Specification, page 11, lines 18 to 21.) In yet another example, "mimic device 20 [could] act on behalf of network messages from external network 10 which are directed to IP addresses assigned to the printers on [a] USB local network." (Specification, page 45, lines 10 to 15).

Mimic device 20 thus acts as a bridge between external network 10 and local network 14 and intercepts messages from client network devices (such as server 11) on the external network which are addressed to legacy or other target devices (such as printer 15) on the local network. Therefore, while a client network device (such as server 11) may send a message addressed to the IP address of a target device (such as printer 15) on the local network, it is mimic device 20 which receives the message.

Once the message is received, mimic device 20 determines if an application module residing in the mimic device is configured to process a functionality requested by

the incoming message. For example, mimic device 20 might include an e-mail printing application module 71 and a secure printing application module 70 as shown in Figure 3.

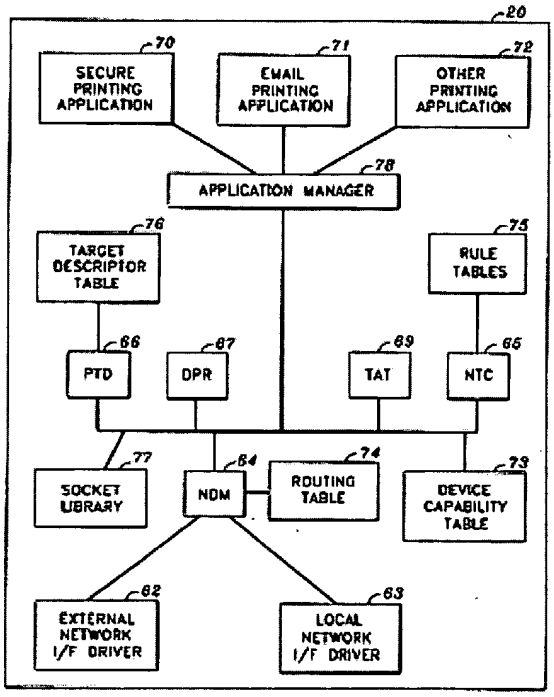

FIG. 3

Mimic device 20 determines if one or more of those modules is configured to process the functionality requested by a message. (See Specification, page 4, lines 15 to 21, page 26, line 24 to page 27, line 13 and page 30, line 1 to page 31, line 32). Thus, the invention determines if an application module in mimic device 20 can process a requested functionality, and it makes this determination independently of whether the target device can or can not also provide this functionality. (See, e.g., Specification, page 4, lines 19 to 21.)

If one of the application modules in mimic device 20 is configured to process the functionality requested by the incoming message, mimic device 20 redirects the message to the application module. (See Specification, page 7, lines 1 to 7 and page 38,

lines 1 to 27). Thus, mimic device 20 can essentially override outdated functionality in a target device, since any message which an application module is configured to process is redirected to the application module, regardless of the capabilities of the target device. Accordingly, the functionality of the target devices can be augmented repeatedly by simply upgrading the application modules in mimic device 20, rather than updating or upgrading each individual target device on the local network. (See, e.g., Specification, page 44, lines 8 to 22).

On the other hand, if the application module or modules is not configured to processed the functionality in the incoming message, mimic device 20 passes the incoming message through the local network to the target network device residing on the local network. (See Specification, page 37, line 32 to page 38, line 27).

By virtue of the foregoing arrangement, the mimic device 20 may act as a "middle man" to augment the functional capabilities of legacy devices or other target devices on a network which lack a desired functionality, ordinarily without the need to add potentially expensive or inefficient software or hardware upgrades to each individual device.

(6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether the rejection of Claim 1 under 35 U.S.C. § 103(a) over Sugiura in view of Cooper should be reversed.

2. Whether the rejection of Claim 33 under 35 U.S.C. § 103(a) over Sugiura and Cooper in view of Banginwar should be reversed.

(7) ARGUMENT

For the procedural purposes of this Appeal only, Claim 1 and the claims dependent from Claim 1 rise and fall together, and Claim 33 and the claims dependent from Claim 33 rise and fall together.

These claims are being argued together solely for the procedural purposes of expediting the consideration and processing of this Appeal, and without conceding the separate consideration of these claims for any other purpose, such as in litigation.

The Rejection of Claim 1 under 35 U.S.C. § 103(a) over Sugiura in View of Cooper

Claims 1 to 32, 34 to 37 and 39

Sugiura is directed to a print control method in which a computer of a LAN transmits print data to a printer of another LAN via the Internet. On the Internet, a terminal device transmits print data to a print server managing a printer, so as to use the printer for printing. HTTP is used as a communication protocol between the terminal device and the print server. In the terminal device, a header including information about the printer to be used is added to the print data, which are transmitted. In the print server, the print data are received, and the print data are transmitted to the printer in accordance with the header of the print data. (See Sugiura, Abstract).

The Examiner contends that Sugiura's print server 23 corresponds to a computing device that performs the step of receiving an incoming message from a client

network device on an external network, wherein the message is addressed to the IP address of a target network device on a local network.

As understood by Appellants, Sugiura's print process begins when a user of a terminal device such as device 33 designates print data DT1 to be printed and a printer PT to be used. (See Sugiura, Figure 9 and paragraph 0106). A header DTa including information about the address of the designated printer is then added to the print data DT1 so as to generate HTTP data DT. (See Sugiura, paragraph 0108). Specifically,

> "The HTTP generating portion 333 [of terminal device 33] generates the HTTP data DT by adding a header to the print data DT1 as shown in FIG. 8A. The header DTa includes information about the address of the printer PT to be used for printing." (Sugiura, paragraph 0095).

Next, the HTTP data DT is transmitted from terminal device 33 to the print server. (See Sugiura, Figure 9 and paragraphs 0108, 0109 and 0116). The print server uses a CGI program to acquire the HTTP data DT, and to understand the nature of the request. Specifically, print server 23 acquires the HTTP data DT from terminal device 33 when a user designates the CGI program in print server 23. In particular,

> "The HTTP data acquiring portion 231 [of print server 23] receives the HTTP data DT via CGI....[t]he CGI program is activated when the terminal device 33 designates it in such a manner as "yyyyy.com/print.cgi". The code "print.cgi" indicates a program that demands the HTTP data DT from the terminal device 33, acquires the HTTP data DT and information about the address of the printer PT as an output target from the terminal device 33 and transmits the print data DT1 to the printer PT." (Sugiura, paragraphs 0096 and 0097).

Once print server 23 acquires the HTTP data DT from terminal device 33 using the CGI program, it uses the information in the HTTP header to route the print data to the printer. (See, e.g, Sugiura, paragraphs 0014 and 0115). Thus, Sugiura's print server acquires the HTTP data from the terminal device, and transmits the print data to the designated printer in accordance with the information in the header DTa. (See Sugiura, Abstract and paragraphs 0020 and 0124).

Accordingly, Sugiura's print data is "addressed" to the print server, and specifically the CGI program of the print server (i.e., "yyyyy.com."), rather than being addressed to the network address of the printer itself. (See, e.g., Sugiura, paragraph 0116).

In another embodiment, Sugiura's print server 23 may be combined with an HTTP server 26, which performs the function of acquiring the HTTP data. (See Sugiura, paragraph 0122). However, in this embodiment, the print data is still transmitted to the HTTP server for routing to the printer designated by the header. In particular,

> "[T]he HTTP server receives the HTTP data DT ...[t]hen, the print data DT1 are extracted from the received HTTP data DT, and the print data DT1 are transmitted to the printer designated by the HTTP header DTa of the print data DT1." (Sugiura, paragraph 0124).

Thus, in both cases, Sugiura's HTTP data is addressed to a server, which uses the HTTP header in the data to route the print data to the printer designated by the header. In this way, Sugiura's system may allow a user to print on an internet printer without being blocked by firewalls which restrict communication protocols other than HTTP. (See Sugiura, paragraphs 0012 and 0045).

Cooper, for its part, discloses an intelligent print driver (IPD) and software simulation used to process a print job. Cooper's system receives a call from an application to print a print job to a selected printer, wherein the print job includes a request to use a hardware service. Cooper's IPD provides a print preview, and if an indication to print is received, the print job is sent to a device driver for the selected printer to print the print job. Cooper's IPD then determines whether the selected printer supports the hardware service. If the hardware service is unsupported by the selected printer, the hardware service is instead emulated using a software simulation. (See, e.g., Cooper, Abstract and Column 5, line 48 to Column 6, line 10).

The Examiner contends that Cooper's software simulation corresponds to the claimed application module residing in the computing device, and additionally contends that Cooper performs the step of determining whether an application module is configured to process the functionality requested by a message.

As understood by Appellants, Cooper's print process begins when the user selects Cooper's intelligent print driver (IPD) and intelligent print driver options for printing, such as an "N Up" type printing option (i.e., display of two or more pages on a page). (See Cooper, Column 7, lines 1 to 13). At the IPD end, a determination is made as to whether a print option has been selected for a received print job. (See Cooper, Column 7, lines 29 to 32).

Cooper's process then determines if the print options selected by the user are supported by the hardware of the selected printing platform. (See Cooper, Column 7, lines 34 to 36). In particular,

> "This determination is made by comparing the IPD print properties, such as, for example, N Up, brochure service, or watermark with the printer properties, such as those found within printer properties 406 in FIG. 4, to determine whether support is present within the particular hardware for this selected option." (Cooper, Column 7, lines 36 to 43).

In this regard, "[p]rinter properties 406 contains information as to the properties or functions supported by the particular printer, for example, whether an N Up function is supported by the printer." (Cooper, Column 5, lines 60 to 63). In this way, Cooper's system makes a determination of whether the selected printer supports a hardware service requested by a user. (See, e.g., Cooper, Abstract).

If it is determined that the selected printer does not support the print option selected by the user, the print option is emulated for the selected printer by the software simulation residing in Cooper's intelligent print driver. In particular, "[i]f hardware support is not present for the print option in step 612, then the device independent format information is retrieved (step 614) and a software simulation is performed (step 616). The software simulation is used to provide the print option not supported by the hardware." (Cooper, Column 7, lines 43 to 48). For example,

> "If the user has selected a print option, such as N Up, the intelligent print driver will query the actual platform driver to see if the platform driver supports N Up. If such support is not found, then the input intelligent print driver will simulate the N Up feature via common scaling and translation techniques." (Cooper, Column 7, line 63 to Column 8, line 1).

Once all of the unsupported selected print options have been simulated by Cooper's software simulation, the process sends the print job to the selected platform for printing the print job. (See Cooper, Column 7, lines 53 to 56).

In this way, Cooper " implement[s] software simulation when selected hardware does not support a particular printing-feature." (Cooper, Column 8, lines 3 to 7).

I. CLAIM 1 IS NOT OBVIOUS OVER SUGIURA IN VIEW OF COOPER BECAUSE SUGIURA AND COOPER FAIL TO TEACH OR SUGGEST AT LEAST FOUR CLAIM LIMITATIONS OF CLAIM 1.

Consistent with the case law, MPEP § 2143 provides that in order to establish a *prima facie* case of obviousness, three basic criteria must be met: there must be motivation to modify the reference or combine reference teachings, there must be a reasonable expectation of success, and the prior art reference (or references when combined) must teach or suggest all of the claim limitations.

In the rejection of Claim 1, there are at least four instances where the rejection omits details of explicitly claimed limitations, or addresses functionality completely different from the plain meaning recited by the words in the claim. Moreover, the applied art simply fails to disclose any of these four claimed features. Finally, there is no motivation to combine the Sugiura and Cooper references. Accordingly, the § 103(a) rejection is deficient and should be reversed. For purposes of readability, each specific claim limitation at issue will be discussed separately.

A. **Claimed Feature: "receiving an incoming message from a client network device residing on the external network, the incoming message being addressed to a network address of a target network device residing on the local network"**

1. The Applied Art Fails to Disclose the Claimed "Receiving" Step Because the Messages in Sugiura Are Addressed to Print Server 23, And Not to a Target Device

Sugiura is directed to a print control method in which a computer of a LAN transmits print data to a printer of another LAN via the Internet. On the Internet, a terminal device transmits print data addressed to a print server managing a printer, so as to use the printer for printing. In the terminal device, an HTTP header with information about the printer to be used is added to the print data, which are transmitted. In the print server, the print data are received, and the print data are transmitted to the printer in accordance with the header of the print data. (See Sugiura, Abstract).

Specifically, Sugiura's print data includes an attached HTTP header containing information about the address of a printer to be used. However, it is Sugiura's print server to which the print data is addressed. A user designates the URL of a CGI program used by the network address of the print server, such as "yyyyy.com/print.cgi". (See Sugiura, Figure 9 and paragraphs 0095 to 0097 and 0114). In particular, as discussed above, print server 23 acquires the HTTP data DT from terminal device 33 when a user designates the CGI program in print server 23, and print server 23 uses the information in the HTTP header to route the print data to the printer. (See Sugiura, paragraphs 0096, 0097 and 0116).

Thus, Sugiura's print data is "addressed" to the print server (i.e., "yyyyy.com."), and the payload in the HTTP header (i.e., "print.cgi") designates a CGI

program executable by the print server. The print data is not addressed to the network address of the printer, which is instead identified by the information in the HTTP header. Sugiura's server receives the data using the CGI program, removes the HTTP header, and transmits the print data to the designated printer in accordance with the information in the header. (See Sugiura, Figures 2 and 11 and paragraphs 0020, 0095 to 0098 and 0124). Specifically,

> "The HTTP data acquiring portion 231 [of print server 23] receives the HTTP data DT via CGI....[t]he CGI program is activated when the terminal device 33 designates it in such a manner as "yyyyy.com/print.cgi". The code "print.cgi" indicates a program that demands the HTTP data DT from the terminal device 33, acquires the HTTP data DT and information about the address of the printer PT as an output target from the terminal device 33 and transmits the print data DT1 to the printer PT." (Sugiura, paragraphs 0096 and 0097).

Accordingly, Sugiura's incoming message is clearly not addressed to a target network device on a local network, and is instead addressed to the print server. Thus, while Sugiura's incoming message *contains* an address of a target printer in the HTTP header, the message itself is not *addressed* to the target printer. (See Sugiura, Abstract and Figure 11). The print server, and specifically the HTTP data acquiring portion thereof, subsequently uses the HTTP header to transmit the message to the target printer. (See Sugiura, Abstract, Figures 2 and 11 and paragraphs 0057, 0095 to 0097 and 108 to 111). Even in an embodiment where the HTTP acquiring portion is located in an HTTP server rather than the print server, the message is still transmitted to the HTTP

server for routing to the target printer, rather than being addressed directly to the target printer. (See Sugiura, paragraphs 0122 to 0124).

In fact, one of Sugiura's purported advantages is that it provides a method for internet printing to printers on a LAN while *avoiding* addressing printers using communication protocols such as IP. In particular, as disclosed in Sugiura, firewalls commonly block communication protocols other than HTTP, for example by "restricting the communication in accordance with such an IP address". (See Sugiura, paragraphs 0012 and 0044). However, Sugiura notes that HTTP can ordinarily bypass a firewall:

> "In this embodiment, it is supposed that firewalls 22 and 32 are not set about restriction of communication using the HTTP and are set about a certain restriction of communication using other communication protocols...it is normal that a firewall is not set about the restriction of communication using the HTTP so that any user can access Web contents." (Sugiura, paragraph 0045.)

Thus, Sugiura's system is actually intended to *avoid* sending a message addressed to a specific network address of a target printer, and rather transmits HTTP data to a print server to route the print data to the printer, so as to bypass firewalls. (See Sugiura, paragraph 0113).

In contrast, in embodiments of the Claim 1 invention, the incoming message received by mimic device 20 is actually addressed to an IP address of the target device, for example the IP address "85.210.1.12." shown as the "Destination IP" of an incoming message in Figure 8A. (See, e.g., Figure 8A).

In summary, Sugiura's messages are not addressed to the target device as claimed by Appellants, but rather are addressed to a server, which may then use the content

of the message to route the message. In particular, Sugiura's print server uses the content of a message, and specifically the HTTP header, to route the message to a target device. Accordingly, while Sugiura's messages may be routed to a logical endpoint, the message is addressed to a server and not to the endpoint.

Thus, Sugiura fails to disclose or suggest the feature of receiving an incoming message from a client network device residing on an external network, the incoming message being addressed to a network address of a target network device residing on a local network. Therefore, the applied art fails to disclose this limitation of the invention, and the rejection of Claim 1 should be reversed.

2. *The Examiner's Rejection Does Not Meet the Language of The Claimed "Receiving" Step.*

MPEP 2143.03 provides that "all words in a claim must be considered in judging the patentability of that claim against the prior art."

The Examiner's Final Rejection of Claim 1 asserts that Sugiura discloses "receiving an incoming message from a client network device residing on the external network, the message being addressed to a target network device." Important claim limitations are missing from this rejection. Of particular note, the Examiner does not address the fact that Claim 1 receives a message addressed to *a network address* of a target network device, an omission which underscores the Examiner's continued misrepresentation of this claim language.

In particular, the Examiner conflates a message containing an address of a target device with a message itself being addressed to the network address of the target device.

In the Advisory Action dated June 27, 2006, the Examiner explains his rejection as follows: "since [Sugiura's] HTTP data DT contains the address of the target printer in the header of the HTTP data DT, certainly the HTTP data DT can be consider as being addressed to a target device." (Advisory Action, page 2). Again, this ignores the clear language of the claims, which specifies that the message is addressed to the "network address" of the target device. Put another way, the claim language does not simply require a message that is "addressed to a target device", as incorrectly stated by the Examiner; rather, the claim language requires a message that is addressed to the "network address" of the target device.

This omission, and the Examiner's explanation of his rejection, evidence the Examiner's continued indifference to actual claim language. Appellants have repeatedly explained that a message *containing* an address of a target device, as in Sugiura, is not the same as the message itself being *addressed* to the network address of a target device, as in the present invention. Sugiura's HTTP data is addressed to a print server, not to a target printer; it is the print server which then routes data to a target printer. The Examiner's rejection is akin to saying that an envelope addressed to person A, containing a letter with person B's address, is actually addressed to person B. This interpretation is incorrect and conflates two distinct functionalities. Nonetheless, the Examiner's rejection does not address this issue.

Therefore, the rejection does not meet the limitations of Claim 1's "receiving step", nor address all of the words in the limitation. Accordingly, the rejection of Claim 1 should be reversed for this reason as well.

B. Claimed Feature: "determining if an application module residing in the computing device is configured to process a functionality requested by the incoming message"

1. The Applied Art Fails to Disclose the Claimed "Determining" Step Because Cooper Determines The Capabilities of his Target Device, Not an Application Module, and Because It Would Not Have Been Obvious To Modify Cooper to Determine Capabilities of an Application Module.

Cooper discloses a method for processing a print job. A call is received from an application to print the print job to a selected printer, wherein the print job includes a request to use a hardware service. A print preview is provided, and if an indication to print is received, the print job is sent to a device driver for the selected printer to print the print job. A determination is made as to whether *the selected printer* supports the hardware service. If the hardware service is unsupported by the selected printer, the hardware service is emulated using a software simulation. (See Cooper, Abstract) (emphasis added).

The Final Rejection asserts that Cooper (Column 6, lines 23 to 34 and Column 7, lines 29 to 62) discloses Appellants' "determining" step. The cited portions of Cooper refer to Cooper's Intelligent Print Driver (IPD), which provides the software simulation that the rejection equates with the claimed "application module". (See, e.g., Advisory Action, page 2).

However, as stated above, Cooper's Intelligent Print Driver (IPD) does not determine whether a requested hardware service can be performed by the IPD's software simulation. Rather, the determination is whether the requested hardware service can be performed by the selected printer. (See Cooper, Abstract, Column 1, lines 59 to 61 and Column 6, lines 23 to 26). Specifically,

> "This determination is made by comparing the IPD print properties, such as, for example, N Up, brochure service, or watermark with the printer properties, such as those found within printer properties 406 in FIG. 4, to determine whether support is present within the particular hardware for this selected option." (Cooper, Column 7, lines 36 to 43).

Thus, Cooper determines the capabilities of the printer device using printer properties 406, which " contains information as to the properties or functions supported by the particular printer, for example, whether an N Up function is supported by the printer." (Cooper, Column 5, lines 60 to 63). If hardware support for the service is not found in the printer, then the IPD will emulate the service using a software simulation. (See, e.g., Cooper, Column 7, line 63 to Column 8, line 1).

Therefore, Cooper clearly does not determine whether its "application module" (the software simulation) is configured to process a service requested by a user; rather, the determination is whether a *selected printer* can support a hardware service. In particular, Cooper's process determines if the print options selected by the user are supported by the hardware of the selected printing platform (See Cooper, Column 7, lines 34 to 36). If it is determined that hardware support for the option is not present in the

selected printer, Cooper's software simulation emulates the requested option. (See Cooper, Column 7, lines 43 to 48).

In fact, Cooper never even contemplates that its software simulation might be *incapable* of processing a functionality. Cooper simply states "if hardware support is not present, then the device independent format is retrieved and a software simulation is performed." (See Cooper, Column 7, lines 43 to 47). Thus, it logically follows that there is no determination of "if" Cooper's software simulation (the Examiner's "application module") can process the functionality requested by a message, or a "case" in which it can not; as disclosed by Cooper, the software simulation always provides the missing service. Thus, Cooper assumes that the software simulation can always emulate a service which a printer does not support; as such, there is no need to determine the capabilities of Cooper's software simulation, and it would not be obvious or even apparent to modify Cooper to determine the "capabilities" of its software simulation.

Because of these differences in the functionality of Cooper, a combination of Sugiura with Cooper's system can not operate like the invention of Claim 1, nor provide the advantages of such an arrangement. For example, in Cooper, if both the selected printer and the software simulation are capable of processing a requested service, the selected printer will receive the print job even though the software simulation is also capable of performing the service. In other words, Cooper's software simulation is only provided to emulate a functionality when a selected printer lacks a requested functionality. (See, e.g., Cooper, Abstract.) This is contrary to the effect of the claimed invention. In the invention, if both mimic device 20 and a target device are capable of processing the functionality requested by a message, an application device in mimic device 20 will receive

the job, *even though* the legacy or other network device might also be able to process the functionality. This permits the present invention to upgrade the functionality of target devices by simply upgrading application modules in mimic device 20, ordinarily without the need for upgrading individual target devices.

Thus, Cooper does not disclose the claimed "determining" step of Claim 1, and the rejection of Claim 1 should be reversed for this reason as well.

2. The Examiner's Reasoning Demonstrates That the Rejection and Applied Art Are Directed At Functionality Completely Different From the Claimed "Determining" Step.

In his rejection, the Examiner asserts that Cooper (Column 6, lines 23 to 34 and Column 7, lines 29 to 62) discloses the claimed determining step. However, as noted above, these portions refer to the determination of the capabilities of a selected printer, and not of Cooper's software simulation. As can be seen from the Advisory Action, the Examiner equates the claimed "application module" with Cooper's software simulation, and not with the target printer. Thus, the Examiner's rejection addresses the wrong determination. Cooper's alleged determination of whether a target *printer* can support a requested functionality is irrelevant, as the claim language requires a determination of whether an application module in the computing device is configured to process the functionality.

The Advisory Action also reasons that

> "Cooper's "process sends the print job (passing the message) to the printer if the support for the requested functionality is in the printer. This means software simulation (application module) is not configure [sic, configured] to process the functionality requested." (Advisory Action, page 3).

Clearly this reasoning is logically and technologically flawed. It cannot seriously be argued that sending a print job to a printer "if support for requested functionality is in the printer" might somehow involve a determination of the capabilities of Cooper's software simulation. No such logical link exists, nor has any reasoning for such a link been presented. More generally, the Examiner has never taken the position that Cooper discloses a determination of whether a requested functionality can be processed by its software simulation, and indeed, given the disclosure of Cooper, he could not do so.

Thus, the final rejection incorrectly interprets the functionality of the "determining" step of Claim 1, and the cited portions of Cooper do not disclose or suggest the functionality claimed by Appellants. Accordingly, the rejection of Claim 1 should be reversed for this reason as well.

C. Claimed Feature: "redirecting the incoming message to the application module in the case that the application module is configured to process the functionality"

1. The Applied Art Fails to Disclose the Claimed "Redirecting" Step Because Any Redirection in Cooper Does Not Occur In Response To the Claimed Circumstances.

Cooper additionally fails to disclose the "redirection" step claimed by Appellants.

In this regard, Appellants submit that since Cooper does not disclose the determination step recited by Appellants, it logically follows that Cooper can not disclose any actions based on such a determination, including the claimed redirection.

Again, Cooper's determination is of the capabilities of a printer, and not of the "application module" as characterized by the Examiner. Therefore, Cooper can not disclose redirection based on whether an application module is configured to process functionality requested by a message.

Furthermore, as noted above, Cooper does not even contemplate the possibility that its IPD software simulation might not be configured to emulate a requested hardware service. As such, there are not "cases" in which the IPD can or can not perform a functionality requested by a message.

Accordingly, Cooper does not disclose the claimed "redirection" step of Claim 1, and the rejection of Claim 1 should be reversed for this reason as well.

2. *The Examiner's Reasoning Demonstrates That the Rejection and Applied Art Are Directed At Functionality Completely Different From the Claimed "Redirecting" Step, and, if Correct, Demonstrates that the Applied Art Does Not Disclose the Claimed "Redirecting" Step.*

In his rejection, the Examiner asserts that Cooper (Column 6, lines 23 to 34 and Column 7, lines 29 to 62) discloses the claimed redirection step.

However, as discussed in detail above, redirection in Claim 1 occurs based on the configuration of an application module, and not based on the configuration of a printer as asserted by the Examiner.

Moreover, the Advisory Action incorrectly interprets the claimed circumstances in which redirection occurs. To reiterate, Claim 1 recites "redirecting the incoming message to the application module in the case that the application module *is* configured to process the functionality". However, the Advisory Action states: "...if

support is not present [in Cooper's printer]...the process must redirected [sic, redirect] the print job to the software (application module) that performs the simulation." (Advisory Action, page 2) (emphasis added). Thus, assuming solely for argument that the Examiner is completely correct in his assertion about Cooper, Cooper redirects a message in the case that support for a hardware service is *not* present in a printer, whereas Claim 1 recites redirection of a message in the case that an application module *is* configured to process a functionality.

Thus, the Examiner's rejection incorrectly interprets the redirecting step of Claim 1, and, if correct, demonstrates that Cooper does not disclose the claimed redirecting step. Accordingly, the rejection of Claim 1 should be reversed for this reason as well.

D. Claimed Feature: "passing the incoming message through the local network to the target network device residing on the local network in the case that the application module is not configured to process the functionality"

1. The Applied Art Fails to Disclose the Claimed "Passing" Step Because Any Passing In Cooper Does Not Occur In Response To the Claimed Circumstances.

Cooper also fails to disclose the claimed passing step.

As above, Appellants submit that since Cooper does not disclose the determination step recited in Claim 1, it logically follows that Cooper can not disclose any actions based on such a determination, including the claimed passing step.

Of particular note, the fact that Cooper does not even contemplate the possibility that its IPD is not configured to process a message specifically negates the possibility of the claimed circumstances for passing a message existing in Cooper.

Moreover, as discussed above, Cooper's determination is of the capabilities of the printer, not the "application module" as characterized by the Examiner.

Accordingly, Cooper does not disclose the claimed "passing" step of Claim 1, and the rejection of Claim 1 should be reversed for this reason as well.

2. *The Examiner's Reasoning Demonstrates That the Rejection and Applied Art Are Directed At Functionality Completely Different From the Claimed "Passing" Step, and, if Correct, Demonstrates that the Applied Art Does Not Disclose the Claimed "Passing" Step.*

In his rejection, the Examiner asserts that Cooper (Column 6, lines 23 to 34 and Column 7, lines 29 to 62) discloses the claimed passing step.

However, as discussed in detail above, passing a message in Claim 1 occurs based on the configuration of an application module, and not based on the configuration of a printer as asserted in the Advisory Action.

Moreover, the Advisory Action also incorrectly interprets the claimed circumstances in which passing occurs. Claim 1 recites passing the incoming message through the local network to the target network device residing on the local network in the case that the application module *is not* configured to process the functionality. However, the Advisory Action states that Cooper's "process sends the print job (passing the message) to the printer if the support for the requested functionality *is* in the printer." (Advisory Action, page 3) (emphasis added). Thus, again assuming the Examiner is completely correct in his assertion, Cooper discloses passing a message in the case that a printer *can* process a requested functionality, whereas Claim 1 recites passing a message to a target device if an application module *can not* process a functionality.

Thus, the Examiner's rejection incorrectly interprets the passing step of Claim 1, and, if correct, demonstrates that Cooper does not disclose the claimed passing step. Accordingly, the rejection of Claim 1 should be reversed for this reason as well.

II. CLAIM 1 IS NOT OBVIOUS OVER SUGIURA IN VIEW OF COOPER BECAUSE THERE IS NO MOTIVATION TO COMBINE THE SUGIURA AND COOPER REFERENCES.

Notwithstanding that Sugiura and Cooper fail to disclose at least four claimed limitations of the present invention, Appellants additionally submit that there is no motivation to combine Sugiura and Cooper.

Sugiura describes using a print server to route HTTP print data to a selected printer in accordance with a header attached to the print data, but does not describe using a software simulation to provide print options unsupported by a selected printer. Cooper, on the other hand, describes using a software simulation to provide print options which a selected printer does not support, but does not describe using a print server to route print data to a selected printer using an HTTP header attached to the print data. Appellants therefore respectfully assert that there is a disconnect between these two references, such that there is no motivation to combine them other than an impermissible motivation which is found in Appellants' own specification.

The MPEP and Federal Circuit case law is explicit on the requirements for motivation to combine references: the motivation must come from the references themselves and not from Applicant's disclosure. The MPEP specifically states that "[t]he teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in Applicant's disclosure". (MPEP § 2143).

Moreover, the fact that the references can be physically combined is not sufficient in and of itself to satisfy the requirement for motivation:

> "The mere fact that references can be combined or modified does not render the resulting combination obvious unless the prior art also suggests the desirability of the combination." (*Id* at page 2100-137, emphasis in original.)

Here, the Final Office Action dated March 31, 2006 provided only one single rationalization as to why it might have been obvious to combine Sugiura and Cooper; at pages 3 and 4 of the Office Action, it states that such a combination would have been obvious since

> "Cooper's teaching of determining if an application module residing in the computing device is configured to process a functionality requested by the incoming message would increase the functionality of Sugiura's system by providing a target network device with printer support for a number of different applications and device drivers. (col.1, lines 46-49; col. 2, lines 33-36)."

Columns 1 and 2 of Cooper, relied on as the suggestion for motivation, state:

> "Therefore, it would be advantageous to have an improved method and apparatus for providing printer support for a number of different applications and device drivers. The present invention provides a method, apparatus, and instructions for providing functions, such as, for example, print preview support that is independent of the particular application or printer device driver." (Cooper, Column 1, lines 46 to 49 and Column 2, lines 33 to 36).

Clearly, this is a *post hoc* rationalization, based on Appellants' own disclosure and claims. The cited portions of Cooper simply describe the potential

advantages of Cooper's system *itself*; nowhere is there seen to be any indication to combine Cooper with any other system, much less a system like Sugiura's which provides internet printing using an HTTP header and a print server to bypass a firewall.

Specifically, Sugiura describes using a print server to transmit print data to a selected printer in accordance with a header attached to the print data. Sugiura does not describe using a software simulation to provide print options unsupported by the selected printer, and most certainly does not specify that functionality could be increased by adding printer support "independent of a particular application or device driver", much less simulating printer functions by software.

Conversely, Cooper describes using a software simulation to provide print options which a selected printer does not support, but does not describe using a print server to transmit print data to a selected printer using an HTTP header attached to the print data. As noted above, the cited portions of Cooper simply describe the potential advantages of Cooper's system on its own; nowhere is there seen to be any indication to combine Cooper with a system implementing an HTTP header and a print server for internet printing.

Moreover, while it might generally be an objective to "increase the functionality of Sugiura's system", as asserted by the Examiner, this statement does not explain how such an objective might be attained. Such generalized objectives cannot be the support for the specific combination proposed in the Office Action. Certainly, a generalized objective of "increasing functionality" does not lead to the conclusion that a method of printing data to a printer of an LAN using HTTP and a server (per Sugiura) might somehow be increased through the use of a software simulation to provide print options which a selected printer does not support (per Cooper). Again, at the time of the

invention, Sugiura and Cooper were unrelated pieces of art; it was only after examination of Appellants' own disclosure that the Examiner concluded that they might, in fact, be useful together.

There is frankly nothing in either of these references which would commend one to the other, such that those of ordinary skill in the art would have combined the reference as described in the Office Action.

It is therefore respectfully submitted that the Office Action's proposed combination of Sugiura and Cooper is without technological or legal basis, such that the rejection under § 103(a) should be reversed for this reason as well.

The Rejection of Claim 33 under 35 U.S.C. § 103(a) over Sugiura and Cooper in view of Banginwar

Claims 33 to 38 and 40

Independent Claim 33 also generally concerns a method for mimicking network devices, but includes limitations not present in Claim 1. For at least the reasons presented above, the rejection of Claim 33 over Sugiura and Cooper in view of Banginwar is also deficient and should be reversed.

I. **CLAIM 33 IS NOT OBVIOUS OVER SUGIURA IN VIEW OF COOPER AND BANGINWAR BECAUSE SUGIURA, COOPER AND BANGINWAR FAIL TO TEACH OR SUGGEST AT LEAST FOUR CLAIM LIMITATIONS OF CLAIM 33.**

As discussed above, Sugiura fails to disclose or suggest receiving an incoming message from a client network device on an external network, wherein the

message is addressed to a network address of a target network device on a local network. In particular, Sugiura's messages are addressed to the print server, rather than the target printer, and it is the content of these messages that contain the address of the target printer (in an HTTP header).

For the same reason, Sugiura also fails to disclose or suggest Claim 33's feature of receiving an incoming message from a client network device residing on an external network, the incoming message being addressed to an IP address of a designated one of a plurality of target network devices on a local network.

Moreover, as discussed in detail above in regards to Claim 1, Cooper is not seen to disclose or suggest the features of determining if an application module residing in a computing device is configured to process a functionality requested by an incoming message, redirecting the incoming message to the application module in the case that the application module is configured to process the functionality, and passing the incoming message to a target network device residing on the local network in the case that the application module is not configured to process the functionality.

In particular, Cooper's "determination" is of the capabilities of a target printer, not an application module, and Cooper in fact does not contemplate that its software simulation (the Examiner's "application module") could ever be incapable of processing a requested functionality.

Accordingly, Appellants respectfully submit that Cooper also does not disclose or suggest the features of Claim 33, specifically the features of determining if an incoming message requests a functionality that an application module is configured to perform, redirecting, in the case that the incoming message requests a functionality that the

application module is configured to perform, the incoming message to the application module which performs the requested functionality in response to the incoming message, and passing, in the case that the incoming message does not request a functionality that the application module is configured to perform, the incoming message through the local network to the designated printing device.

Banginwar is not seen to remedy the above-noted shortcomings of Sugiura and Cooper.

Therefore, the rejection of independent Claim 33 is deficient, and should be reversed.

II. CLAIM 33 IS NOT OBVIOUS OVER SUGIURA IN VIEW OF COOPER AND BANGINWAR BECAUSE THERE IS NO MOTIVATION TO COMBINE THE SUGIURA AND COOPER AND BANGINWAR REFERENCES.

In its rejection of Claim 33, the Final Office Action dated March 31, 2006 provided the same rationalization to combine Sugiura and Cooper as that set forth in its rejection of Claim 1. As discussed in detail above, this proposed combination of Sugiura and Cooper is without technological or legal basis.

Moreover, Banginwar is not seen to add anything that would provide a suggestion to combine Sugiura and Cooper. Pages 15 and 16 of the Final Office Action state that it would be obvious to add Banginwar to Sugiura and Cooper because

> "Banginwar's teaching of discovering legacy network printers would increase the system alertness of Sugiura's and Cooper's systems by allowing new devices added to the system to be notify [sic, notified] to the user."

However, a generalized objective of "increasing system alertness" by allowing new devices to be notified to a user (per Banginwar) is not seen to provide any suggestion that a method of printing data to a printer of an LAN using HTTP and a server (per Sugiura) might somehow be increased through the use of a software simulation to provide print options which a selected printer does not support (per Cooper).

Accordingly, it is respectfully submitted that the Office Action's proposed combination of Sugiura and Cooper is deficient, and the rejection of Claim 33 under § 103(a) should be reversed for this reason as well.

CONCLUSION

Appellants respectfully submit that the 35 U.S.C. § 103(a) rejections of record are deficient for at least the foregoing reasons. Reversal of the rejections is respectfully requested.

Appellants' undersigned attorney may be reached in our Costa Mesa, California office at (714) 540-8700. All correspondence should continue to be directed to our below-listed address.

Respectfully submitted,

Michael K. O'Neill
Attorney for Appellants
Registration No.: 32,622

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-2200
Facsimile: (212) 218-2200

CLAIMS APPENDIX

1. (Previously Presented) A method for mimicking network devices, the method being performed in a computing device having first and second network interface cards, the first network interface card connecting the computing device to an external network and the second network interface card connecting the computing device to a local network, the method comprising the steps of:

receiving an incoming message from a client network device residing on the external network, the incoming message being addressed to a network address of a target network device residing on the local network;

determining if an application module residing in the computing device is configured to process a functionality requested by the incoming message;

redirecting the incoming message to the application module in the case that the application module is configured to process the functionality; and

passing the incoming message through the local network to the target network device residing on the local network in the case that the application module is not configured to process the functionality.

2. (Previously Presented) A method according to claim 1, wherein in the redirecting step, the processing of the functionality by the application module includes sending a response message from the application module over the external network to the client network device, the response message having a source identification address identical to a source identification address of the target network device.

3. (Previously Presented) A method according to claim 1, wherein in the redirecting step, the processing of the functionality by the application module includes sending a local message from the application module over the local network to the target network device which performs a function in response to the local message.

4. (Previously Presented) A method according to claim 1, wherein in the determining step, an inbound rules table is used to determine if the functionality is to be processed by an application module residing in the computing device.

5. (Previously Presented) A method according to claim 4, wherein the inbound rules table contains a plurality of rules, each rule corresponding to one of a plurality of target network devices on the local network.

6. (Previously Presented) A method according to claim 5, further comprising the step of discovering each of the plurality of target network devices on the local network by listening to the local network for messages from the target network devices, creating a target descriptor entry corresponding to each discovered target network device in a target descriptor table, and creating a rule corresponding to each target descriptor entry in the inbound rules table.

7. (Previously Presented) A method according to claim 6, wherein the inbound rules table contains at least one rule which indicates whether a functionality

requested for a corresponding target network device to perform is to be processed by an application module residing in the computing device.

8. (Previously Presented) A method according to claim 7, wherein in the determining step, the incoming message is applied to the plurality of rules in the inbound rules table to determine if the functionality is to be processed by an application module residing in the computing device.

9. (Previously Presented) A method according to claim 7, wherein each rule contains an IP address of the target network device corresponding to the rule to indicate whether a functionality requested for the corresponding target network device to perform is to be processed by an application module residing in the computing device.

10. (Previously Presented) A method according to claim 7, wherein each rule contains a port identifier to indicate whether a functionality requested of the target network device corresponding to the rule is to be processed by an application module residing in the computing device.

11. (Previously Presented) A method according to claim 6, wherein the discovering step includes sending a discovery message to each discovered target network device and receiving discovery information in response to the discovery message from the corresponding target network device, wherein the discovery information is placed in the target descriptor entry for the corresponding target network device.

12. (Previously Presented) A method according to claim 11, further comprising a polling step of sending a discovery message on a periodic basis to each discovered target network device, and receiving in response to the discovery message discovery information from the corresponding target network device, wherein the target descriptor entry is updated with the newly received discovery information.

13. (Previously Presented) A method according to claim 12, wherein in the case that discovery information is not received in response to the discovery message for a particular one of the discovered target network devices, the target descriptor entry corresponding to the particular discovered target network device is deleted.

14. (Previously Presented) A method according to claim 6, further comprising the step of sending a notification to the application module for each discovered target network device, the notification containing information related to the target descriptor entry for the corresponding target network device.

15. (Original) A method according to claim 6, further comprising the step of publishing each target descriptor entry to the application module.

16. (Original) A method according to claim 3, wherein the second network interface card is assigned a preset IP address, and the local message contains a source IP address which is identical to the preset IP address.

17. (Original) A method according to claim 3, wherein the local message contains a source IP address which is identical to a source IP address of the client network device.

18. (Original) A method according to claim 3, wherein the local message contains a source IP address which is identical to a source IP address of the second network interface card.

19. (Previously Presented) A method according to claim 1, wherein in the redirecting step, the processing of the functionality by the application module includes preparation of an outbound message for delivery to a designated device on one of the external network and the local network, and a routing table is used to determine which one of the external network and the local network is used for sending the outbound message to the designated device.

20. (Previously Presented) A method according to claim 19, wherein the routing table contains a cross-reference indicator for each target network device to indicate which one of the external network and the local network is used for sending the outbound message to the designated device.

21. (Original) A method according to claim 19, wherein the routing table is used to determine whether a preset IP address of the second network interface card or a source IP address of the client network device is used as a source IP address in the outbound message.

22. (Original) A method according to claim 7, further comprising the step of tracking a port identifier of a port opened by the application module and creating a rule in the inbound rules table corresponding to the port identifier, wherein in the determining step, the rule is used to redirect a message from the external network to the application module if the message contains the port identifier corresponding to the rule.

23. (Previously Presented) A method according to claim 22, further comprising the steps of tracking an initial target port identifier of a port opened by a target network device, mapping the initial target port identifier to a new target port identifier, creating a first map rule in the inbound rules table corresponding to the target network device which maps the initial target port identifier to the new target port identifier, and creating a second map rule in an outbound rules table corresponding to the target network device which maps the new target port identifier to the initial target port identifier.

24. (Previously Presented) A method according to claim 4, wherein the local network is a USB network, the target network device is a printer, and the inbound rules table contains rules which are used in the determining step to redirect an incoming message for the printer from the external network to the application module which sends a USB message over the local network to the printer in response to the incoming message.

25. (Previously Presented) A method according to claim 4, wherein the local network is a USB network, the target network device is a digital camera, and further including the steps of downloading a digital image to the application module from the

digital camera via the local network, and sending the digital image to a server on the external network.

26. (Previously Presented) A method according to claim 4, wherein the inbound rules table contains rules which are used in the determining step to route an incoming message from the external network to a network device on the local network.

27. (Original) A method according to claim 4, wherein the inbound rules table contains rules which are used in the determining step to capture an incoming message from the external network and further including the step of preventing transmission of the incoming message on the local network.

28. (Previously Presented) A method according to claim 4, wherein the inbound rules table contains rules which are used in the determining step to determine that all incoming messages from the external network are not to be processed by the application module, whereby all incoming messages from the external network are passed through the local network.

29. (Previously Presented) A method according to claim 4, wherein the application module is a file server which sends at least one file over the local network to the target network device and at least one file over the external network to the client network device.

30. (Original) A method according to claim 4, wherein the inbound rules table contains rules which are used in the determining step to determine that a set of designated incoming messages are copied to the application module which records each of the set of designated incoming messages.

31. (Previously Presented) A method according to claim 4, wherein the inbound rules table contains rules which are used in the determining step to detect if the incoming message is an undesirable message, and in the case that the incoming message is an undesirable message, determining that the incoming message is to be processed by the application module, whereby the incoming message is redirected to the application module.

32. (Original) A method according to claim 1, further including the step of transmitting a plurality of undesirable messages from the application module over one of the external network and the local network.

33. (Previously Presented) A method for mimicking network devices, the method being performed in a computing device having first and second network interface cards, the first network interface card connecting the computing device to an external network and the second network interface card connecting the computing device to a local network, the method comprising the steps of:

discovering a plurality of target network devices on the local network by detecting messages on the local network from each of the plurality of target network devices;

creating a rule in a rules table for each of the discovered target network devices, each rule containing the IP address of the corresponding target network device and indicating whether an application module in the computing device is configured to perform a function on behalf of the corresponding target network device;

receiving an incoming message from a client network device residing on the external network, the incoming message being addressed to an IP address of a designated one of the plurality of target network devices;

determining, based at least in part on the rule corresponding to the designated target network device, if the incoming message requests a functionality that the application module is configured to perform;

redirecting, in the case that the incoming message requests a functionality that the application module is configured to perform, the incoming message to the application module which performs the requested functionality in response to the incoming message; and

passing, in the case that the incoming message does not request a functionality that the application module is configured to perform, the incoming message through the local network to the designated target network device.

34. (Previously Presented) A computing device for mimicking network devices, the computing device having first and second network interface cards, the first network interface card connecting the computing device to an external network and the second network interface card connecting the computing device to a local network, said computing device comprising:

a program memory for storing process steps executable to perform a method according to any of claims 1 to 33 and 37 to 40, and

a processor for executing the process steps stored in said program memory.

35. (Previously Presented) Computer-executable process steps stored on a computer readable medium, said computer-executable process steps for mimicking network devices and for being performed in a computing device having first and second network interface cards, the first network interface card connecting the computing device to an external network and the second network interface card connecting the computing device to a local network, said computer-executable process steps comprising process steps executable to perform a method according to any of claims 1 to 33 and 37 to 40.

36. (Previously Presented) A computer-readable medium which stores computer-executable process steps, the computer-executable process steps to mimic network devices and to be performed in a computing device having first and second network interface cards, the first network interface card connecting the computing device to an external network and the second network interface card connecting the computing device to a local network, said computer-executable process steps comprising process steps executable to perform a method according to any of claims 1 to 33 and 37 to 40.

37. (Previously Presented) A method according to claim 1, wherein the target network device is a legacy network device.

38. (Previously Presented) A method according to claim 33, wherein the target network device is a legacy network device.

39. (Previously Presented) A method according to claim 3, wherein in the passing step, the target network device performs the requested functionality in response to the incoming message received from the computing device.

40. (Previously Presented) A method according to claim 33, wherein in the passing step, the target network device performs the requested functionality in response to the incoming message received from the computing device.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.

CA_MAIN 123703v1